Study Guide 3

Since we can’t create instances of abstract classes, what can we do to create an “instance” of the abstract class?

Slide 3, 4, 5, 6, 7

What is the basis of object oriented programming? What all makes up object oriented programming?

Slide 9, 11

What are the main things that FLTK provides? What is the coordinate layout in FLTK?

Slide 17, 18

What can we do with the Shape class? What do all shapes have in common? How do we draw different shapes (the shape class can possibly draw every known shape)?

Slide 20, 21

What does the UML for the shape class and its child classes look like? Why is draw virtual?

Slide 23

How would we use images in FLTK? How would we load an image in FLTK? How does FLTK indicate there was an error loading an image? What is required from the filename to actually make the image load function work?

Slide 27, 29, 30

What does the FL\_Widget::image(FL\_Image); function do? What function sets the image label for when the widget is inactive?

Slide 31

What is event driven programming? What is a widget? What happens when events occur in regards to widgets? What are some examples of events?

Slide 34, 35, 36, 37

How would we start to process events? What is the first way we can get events and what do we do with these events? What is the second way to get events and what does this process entail?

Slide 38, 39, 40

How do we start a GUI program? Describe the typical process behind a GUI program.

Slide 42, 43

What is the process for using handles in a program? What is the handling order of events?

Slide 47,48

What happens when we write our own handle function? Do we ever need to explicitly call handle? What do we return if an event was “handeled,,” and what do we return if it wasn’t? What is the syntax for writing a handle function?

Slide 49, 50

How does an event handler work in FLTK?

Slide 52

What is a callback? What happens with a callback in event driven programming? How many callback functions can a widget register? What would you do if needed more callbacks?

Slide 53, 54

What is the syntax for using the callback function? How do you pass a function as an argument? What is a void pointer and how does it work?

Slide 55, 56

What is syntax for defining a callback? What arguments are passed into the callback function? What can the second argument be (usually) and why is it this? How would we write a function that has 2 widgets passed into it?

Slide 58, 59, 60, 61, 62, 63

What happens when you define a handle() function for a derived class? How would we fix this and where would we put this fix? What happens if we don’t fix this?

Slide 64

What must a callback be if it is class member?

Slide 65

What is an Abstract Data type? What do these focus on? What is an example of an ADT? What are the parts of this example? What is a Queue?

Slide 68, 69, 70, 71

Is a vector an abstract data type? What is a data structure? Does the data structure have to be specified with an ADT? What is an ADT generally in C++ and what is it made up of?

Slide 72, 73, 74